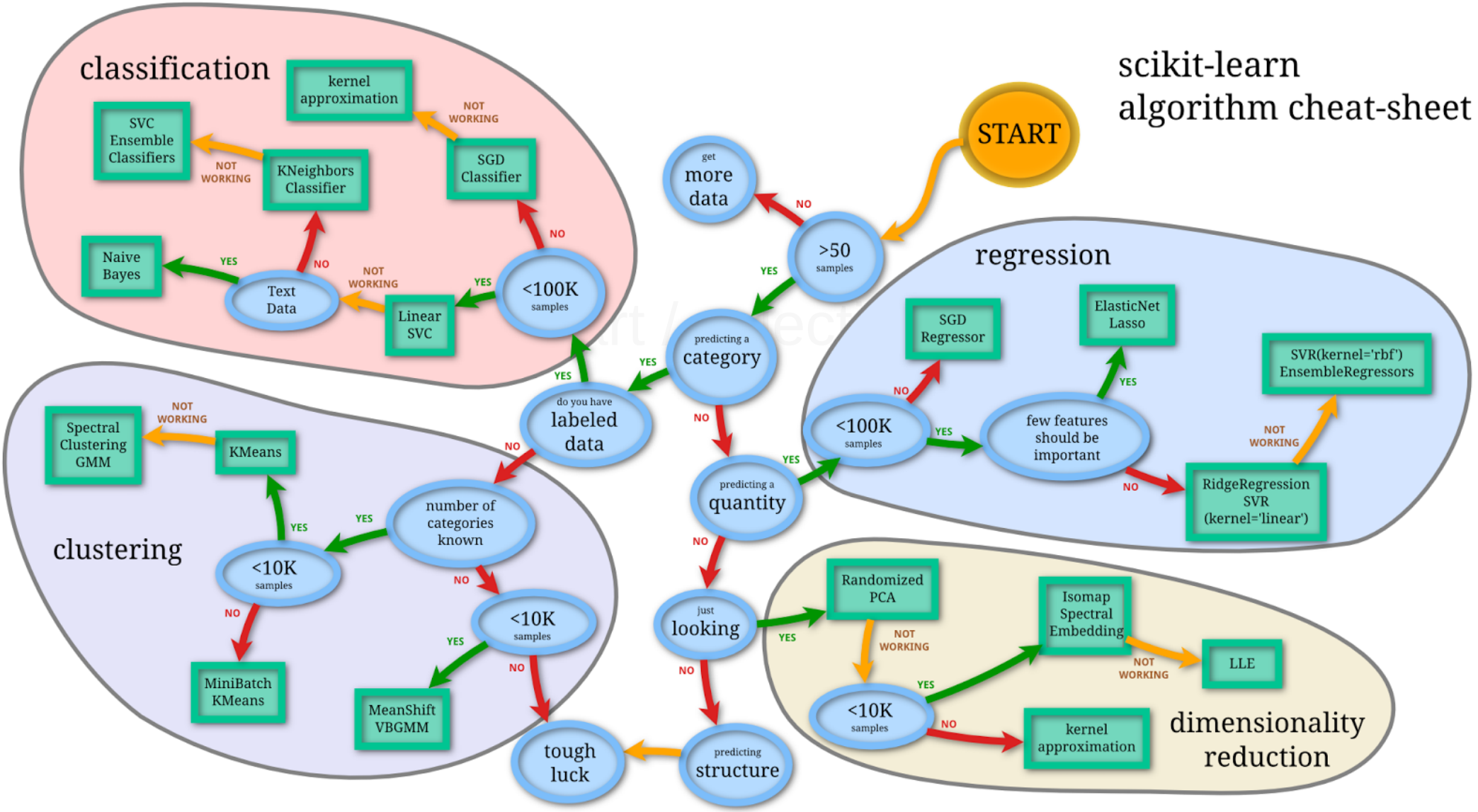
Automatic Machine Learning?

Andreas Müller

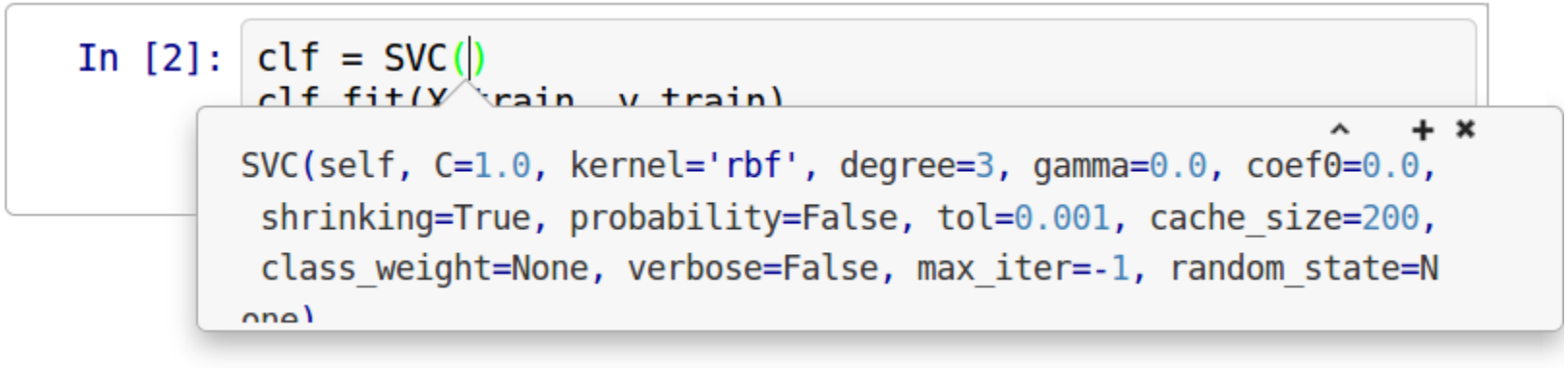
Columbia University, scikit-learn

Why?

Issues with current tools (scikit-learn)


scikit-learn
algorithm cheat-sheet
START
get
more
data
NO
>50
samples
YES
predicting a
category
YES
NO
do you have
labeled
data
YES
NO
predicting a
quantity
YES
NO
just
looking
NO
predicting
structure
tough
luck
classification
kernel
approximation
NOT
WORKING
SVC
Ensemble
Classifiers
NOT
WORKING
KNeighbors
Classifier
SGD
Classifier
NO
Naive
Bayes
YES
Text
Data
NO
NOT
WORKING
Linear
SVC
YES
<100K
samples
YES
regression
SGD
Regressor
ElasticNet
Lasso
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K
samples
YES
few features
should be
important
YES
NOT
WORKING
NO
RidgeRegression
SVR
(kernel='linear')
clustering
Spectral
Clustering
GMM
NOT
WORKING
KMeans
YES
number of
categories
known
YES
NO
<10K
samples
NO
MiniBatch
KMeans
<10K
samples
YES
NO
MeanShift
VBGMM
dimensionality
reduction
Randomized
PCA
YES
NOT
WORKING
Isomap
Spectral
Embedding
NOT
WORKING
LLE
YES
<10K
samples
NO
kernel
approximation

Selecting Hyper-Parameters



Scikit-learn: Explicit is better than implicit

```
make_pipeline(
    OneHotEncoder(),
    Imputer(),
    StandardScaler(),
    SVC())
```

What?

```
from automl import AutoClassifier
clf = AutoClassifier().fit(X_train, y_train)
```

```
> Current Accuracy: 70% (AUC .65) LinearSVC(C=1), 10sec
> Current Accuracy: 76% (AUC .71) RandomForest(n_estimators=20) 30sec
> Current Accuracy: 80% (AUC .74) RandomForest(n_estimators=500) 30sec
```

Step 1: Automate Parameter Selection

Step 2: Automate Model Selection

Step 3: Automate Pipeline Selection

How?

Formalizing the Search Space

Discrete and Continuous Parameters
Conditional Parameters
Fixed pipeline vs flexible pipeline

Formalizing the Search Space

Discrete and Continuous Parameters
Conditional Parameters
Fixed pipeline vs flexible pipeline

Search Methods

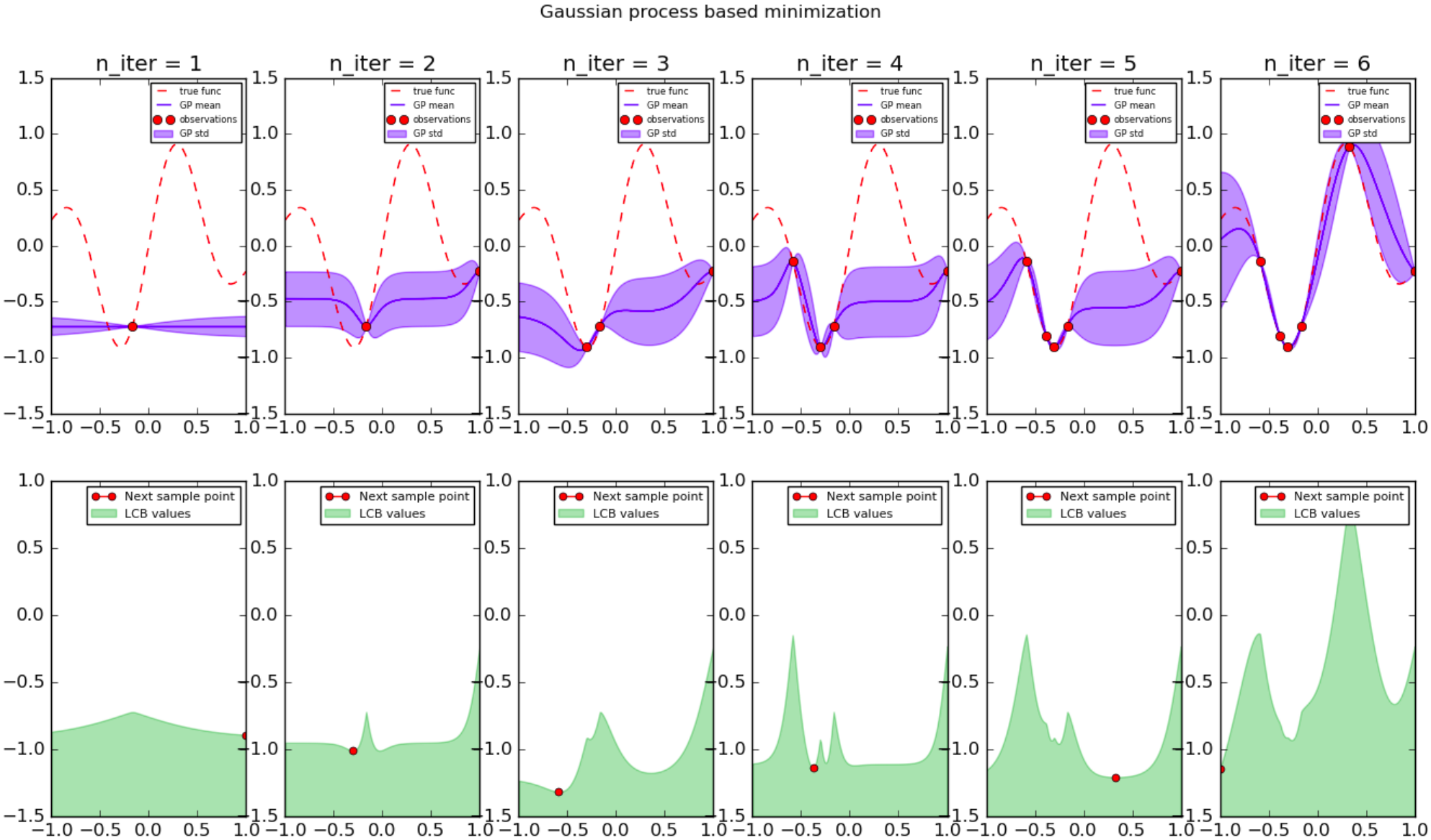
Exhaustive Search (Grid Search)

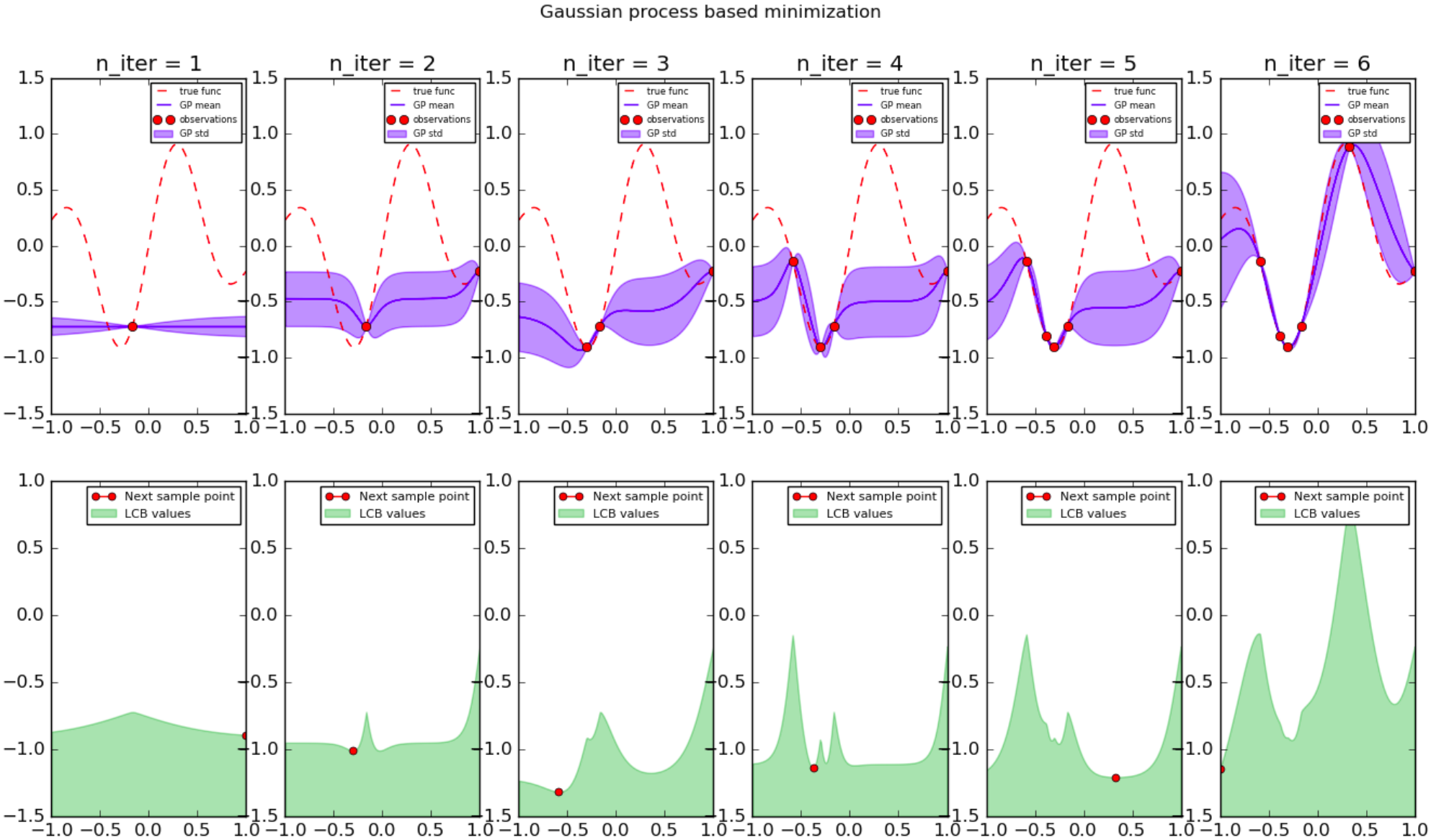
Randomized Search

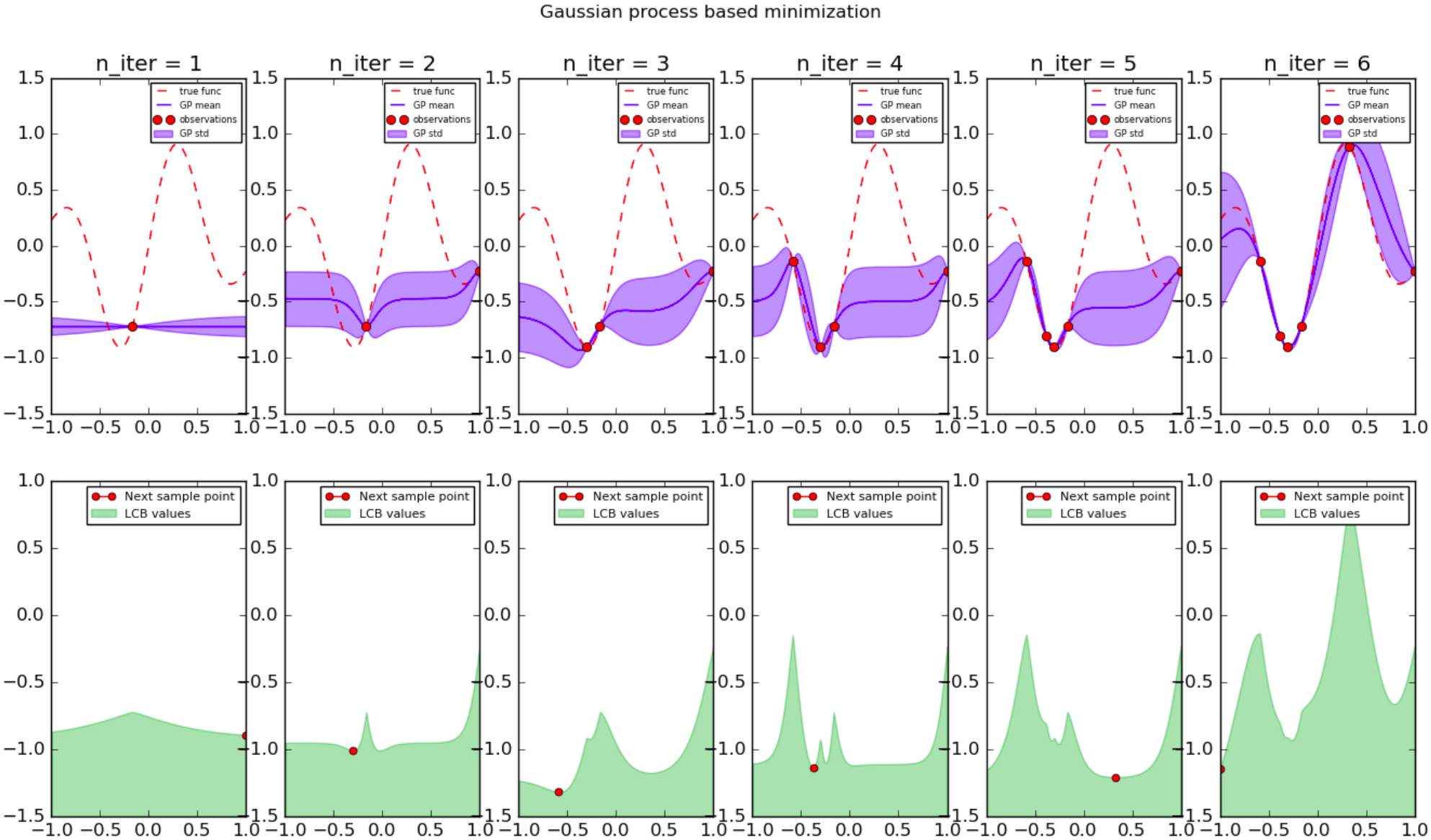
Randomized Search

Bayesian Optimization (SMBO)

Bayesian Optimization (SMBO)


Gaussian process based minimization
n_iter = 1
n_iter = 2
n_iter = 3
n_iter = 4
n_iter = 5
n_iter = 6
true func
GP mean
observations
GP std
Next sample point
LCB values


Gaussian process based minimization
n_iter = 1
n_iter = 2
n_iter = 3
n_iter = 4
n_iter = 5
n_iter = 6
true func
GP mean
observations
GP std
Next sample point
LCB values


Gaussian process based minimization
n_iter = 1
n_iter = 2
n_iter = 3
n_iter = 4
n_iter = 5
n_iter = 6
true func
GP mean
observations
GP std
Next sample point
LCB values

Common Models

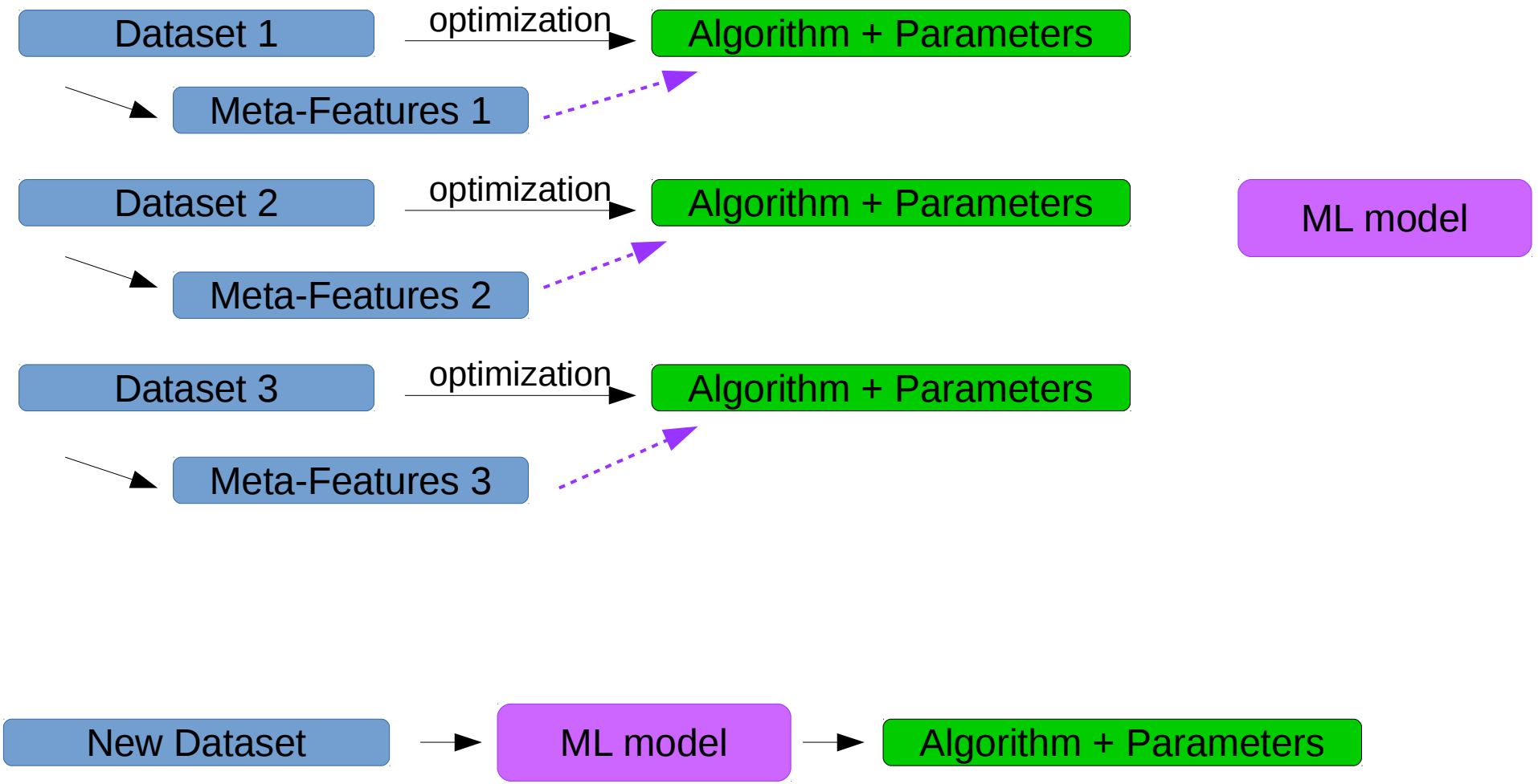
	Discrete Parameters	Scalable	Conditional Parameters	Papers & implementations	Specify parameters
Gaussian Process	?	✗	?	many	bounds
Random Forest (SMAC)	✔	✔	✔	few	prior
Non-parametric (TPE)	✔	?	✔	few	prior

Common Models

	Discrete Parameters	Scalable	Conditional Parameters	Papers & implementations	Specify parameters
Gaussian Process	?	✗	?	many	bounds
Random Forest (SMAC)	✔	✔	✔	few	prior
Non-parametric (TPE)	✔	?	✔	few	prior

Warm-starting and Meta-learning

Meta-Learning



Meta-Features

Existing Approaches

auto-sklearn

(Hutter, Feurer, Eggensperger)

http://automl.github.io/auto-sklearn/stable/

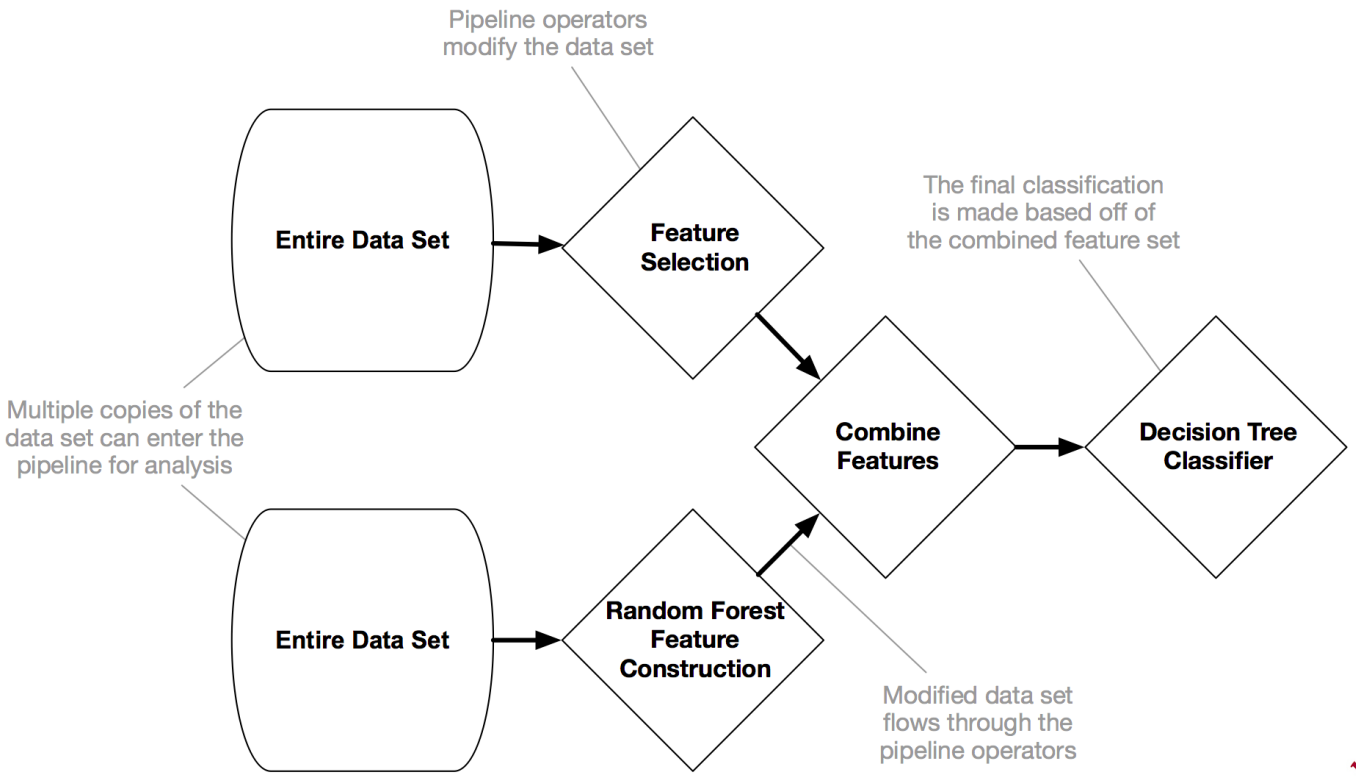
Autoweka

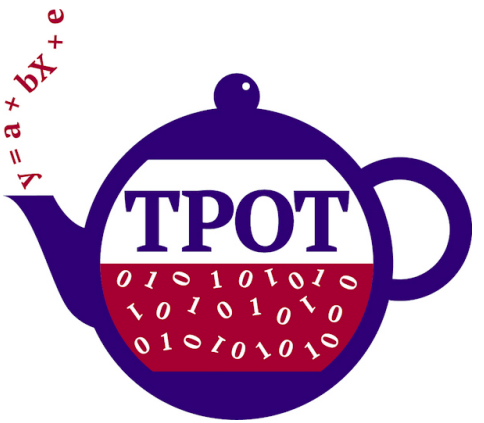
http://www.cs.ubc.ca/labs/beta/Projects/autoweka/

Hyperopt-sklearn

http://hyperopt.github.io/hyperopt-sklearn/

TPot


Pipeline operators
modify the data set
Entire Data Set
Feature
Selection
The final classification
is made based off of
the combined feature set
Multiple copies of the
data set can enter the
pipeline for analysis
Combine
Features
Decision Tree
Classifier
Entire Data Set
Random Forest
Feature
Construction
Modified data set
flows through the
pipeline operators


y = a + bX + e
TPOT

SMBO Packages

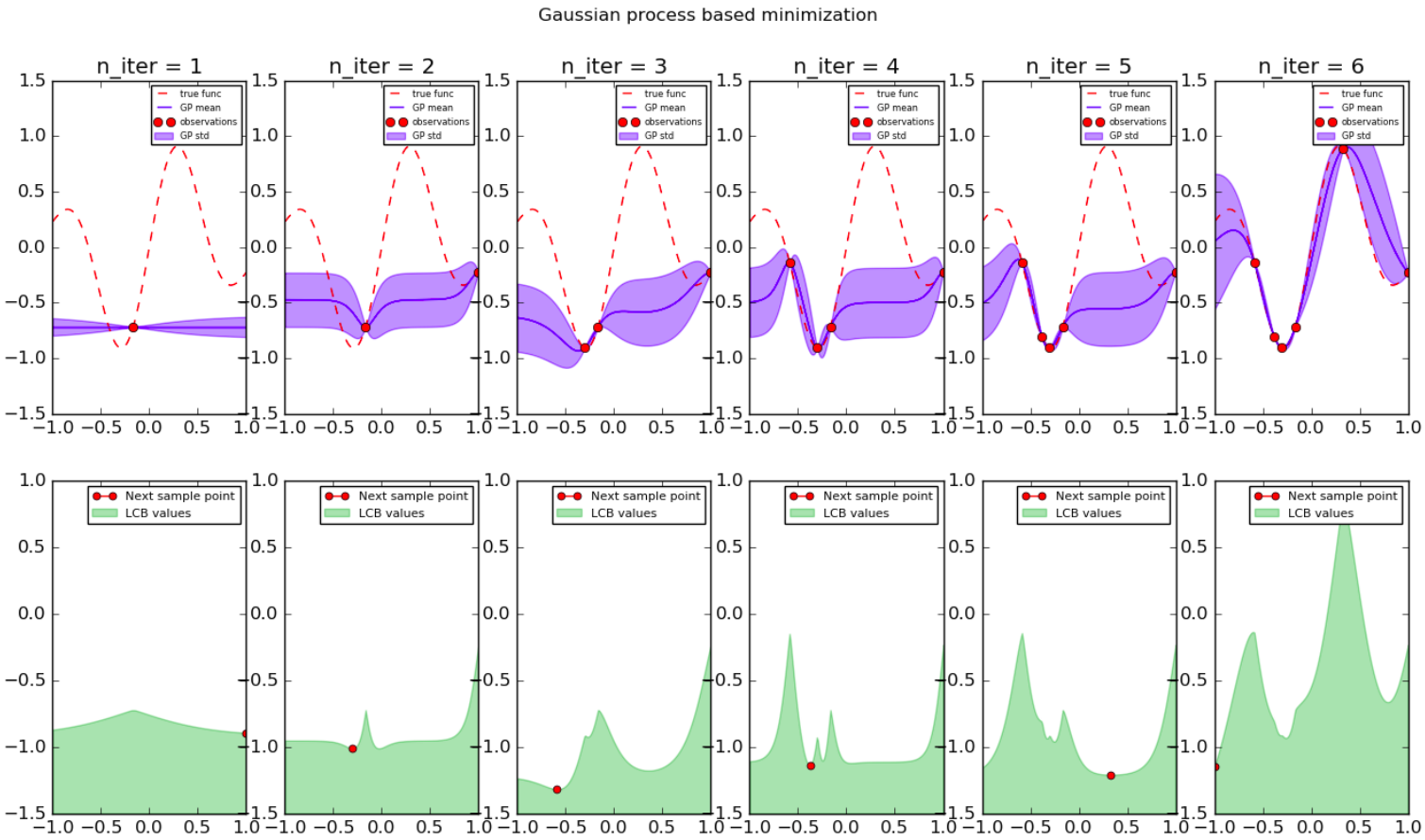
Spearmint

https://github.com/HIPS/Spearmint

GPyOpt

https://github.com/SheffieldML/GPyOpt

Scikit-optimize



Some Benchmarks

Experiment	# Evals	SMAC	TPE	Spearmint	DNGO
Branin (0.398)	200	0.655 ± 0.27	0.526 ± 0.13	$\mathbf{0.398 \pm 0.00}$	$\mathbf{0.398 \pm 0.00}$
Hartmann6 (-3.322)	200	-2.977 ± 0.11	-2.823 ± 0.18	$\mathbf{-3.3166 \pm 0.02}$	-3.319 ± 0.00
Logistic Regression	100	8.6 ± 0.9	8.2 ± 0.6	$\mathbf{6.88 \pm 0.0}$	$\mathbf{6.89 \pm 0.04}$
LDA (On grid)	50	1269.6 ± 2.9	1271.5 ± 3.5	$\mathbf{1266.2 \pm 0.1}$	$\mathbf{1266.2 \pm 0.0}$
SVM (On grid)	100	$\mathbf{24.1 \pm 0.1}$	24.2 ± 0.0	$\mathbf{24.1 \pm 0.1}$	$\mathbf{24.1 \pm 0.1}$

Results below are for `n_calls=64`:

Method	Average rank (less is better)
dummy_minimize	4.552
forest_minimize	2.362
gbrt_minimize	2.172
gp_minimize	1.241
gpyopt_minimize	1.069
hyperopt_minimize	3.052
smac_minimize	3.431

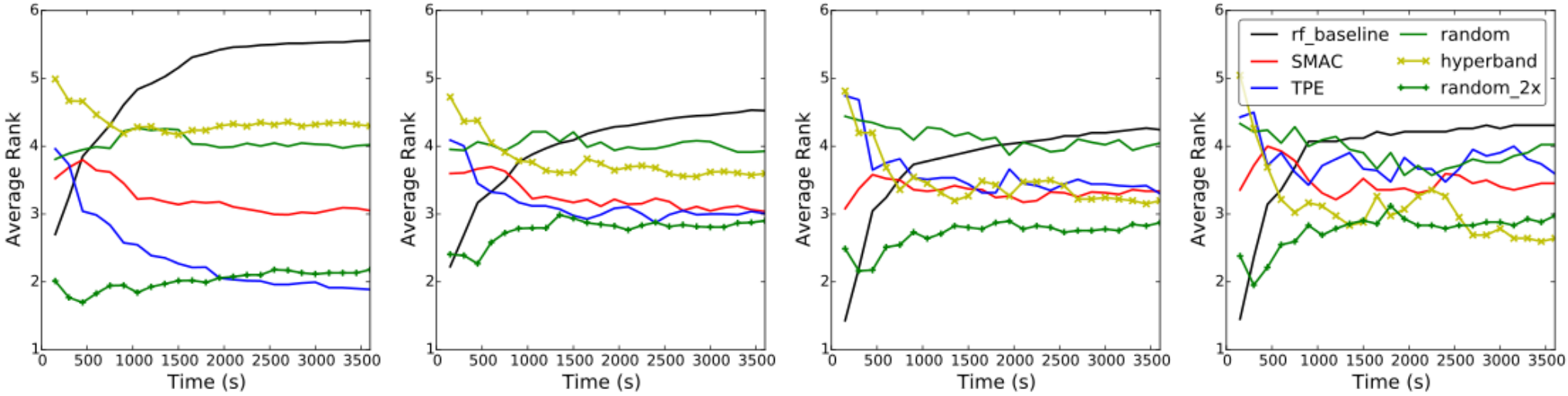
scikit-optimize

GpyOpt (GP)

Hyperopt (TPE)

SMAC (Random Forest)

https://github.com/iaroslav-ai/scikit-optimize-benchmarks


Average Rank
Time (s)
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
rf_baseline
SMAC
TPE
random
hyperband
random_2x

(a) (b) (c) (d)

Within Scikit-learn

- GridSearchCV
- RandomizedSearchCV
- Searching over Pipelines
- Built-in parameter ranges (coming)

TODO

Clean separation of:

- Model Search Space
- Pipeline Search Space
- Optimization Method
- Meta-Learning

- Exploit prior knowledge better!
- Usability
- Runtime consideration

TODO

Clean separation of:

- Model Search Space
- Pipeline Search Space
- Optimization Method
- Meta-Learning

- Exploit prior knowledge better!
- Usability
- Runtime consideration
- Data subsampling

Criticism

Randomized Search works well

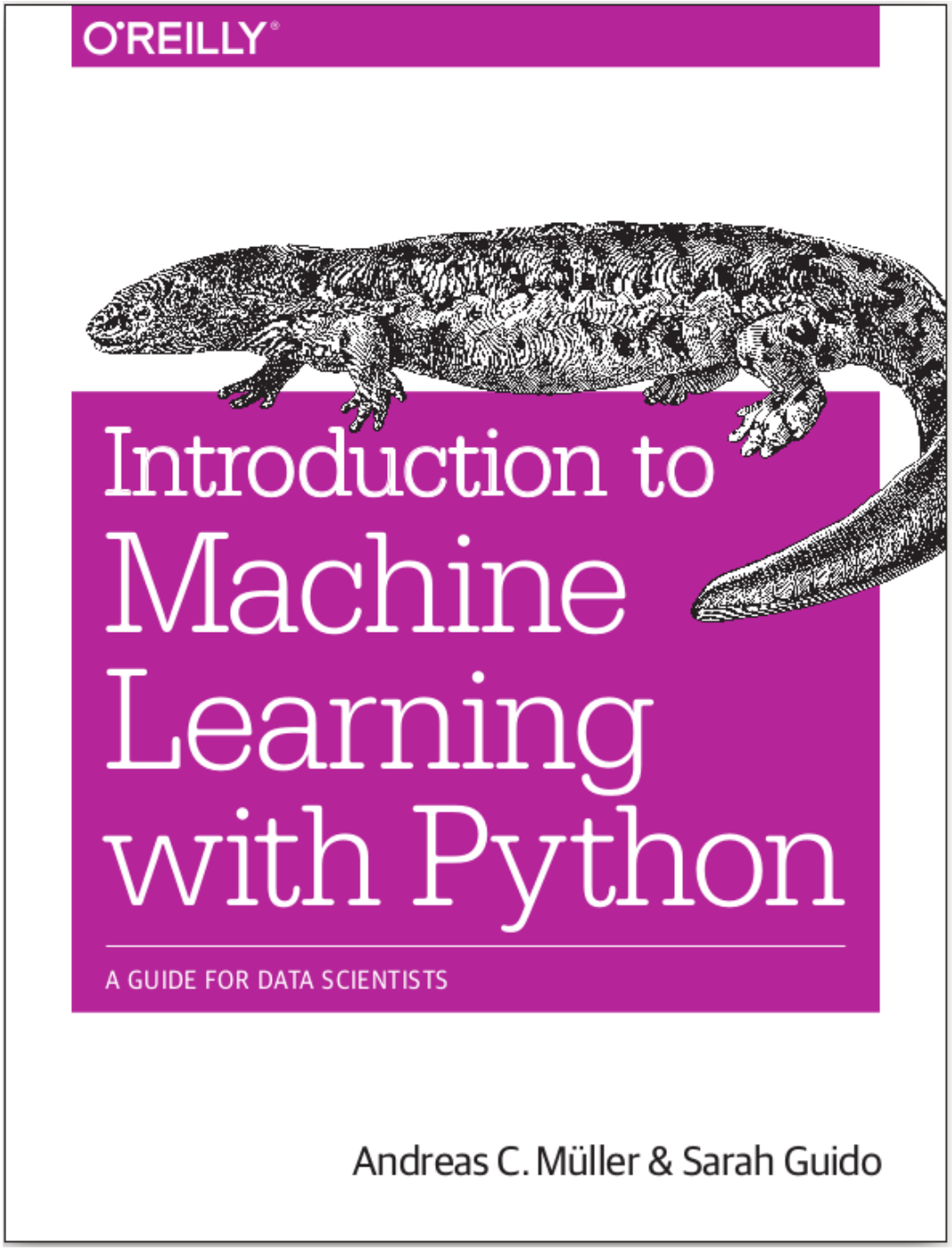
Do we need 100 Classifiers?
Do we need Complex pipelines?

I don’t want a black-box!

Making it too easy?

Material

- Taking the Human Out of the Loop: A Review of Bayesian Optimization (Shahriari, Swersky, Wang, Adams, de Freitas)
- Random Search for Hyper-Parameter Optimization (Bergstra, Bengio)
- Efficient and Robust Automated Machine Learning (Feurer et al) [autosklearn]
- http://automl.github.io/auto-sklearn/stable/
- Efficient Hyperparameter Optimization and Infinitely Many Armed Bandits (Lie et. al) [hyperband] https://arxiv.org/abs/1603.06560
- Scalable Bayesian Optimization Using Deep Neural Networks [Snoek et al]
- https://github.com/iaroslav-ai/scikit-optimize-benchmarks



amueller.github.io

@amuellerml

@amueller

t3kcit@gmail.com

https://github.com/amueller/talks_odt/